

(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10285216 A**

(43) Date of publication of application: 23 . 10 . 98

(51) Int. Cl.

**H04L 12/56**  
**G06F 13/00**

(21) Application number: 10064914

(22) Date of filing: 16 . 03 . 98

(30) Priority: 28 . 03 . 97 US 97 828449

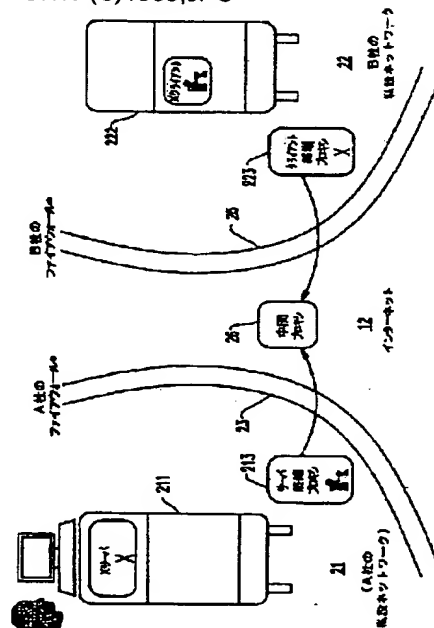
(71) Applicant: **INTERNATL BUSINESS MACH  
CORP <IBM>**(72) Inventor: **CRICHTON JOSEPH M  
GARVIN PETER F  
STATEN JEFFREY W  
WRIGHT WAAIKI L**
**(54) SECURITY PROTECTING COMMUNICATION  
TUNNELLING METHOD AND DEVICE  
THEREFOR**

COPYRIGHT: (C)1998,JPO

(57) Abstract:

**PROBLEM TO BE SOLVED:** To permit a fire wall to execute passing without adding change to a fire wall software and fire wall configuration by using a intermediate server or a proxy.

**SOLUTION:** The fire walls 23 and 25 are provided with a SOCKS function by which an 'inner' client is connected to an 'outer' server. A server terminating proxy 214 is connected to an X-server 211 in the same way as that of an actual X-client. The server terminating proxy 213 is regarded as a local X-client by the X-server 211 so that correction is not required in the X-server 211 in this case. A client terminating proxy 223 and the server terminating proxy 213 use the pre-existed function so as to execute a request from an inner side to an outer side through the fire walls. An intermediate proxy 26 is mutually address possible by the both terminating proxies so that perfect in-terminating end connection is established between the X-client 222 and the X-server 211 with the intermediate proxy 26.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-285216

(43) 公開日 平成10年(1998)10月23日

(51) Int.Cl.<sup>8</sup>

H 0 4 L 12/56

G 0 6 F 13/00

識別記号

3 5 7

F I

H 0 4 L 11/20

G 0 6 F 13/00

1 0 2 Z

3 5 7 Z

審査請求 未請求 請求項の数6 OL (全 19 頁)

(21) 出願番号 特願平10-64914

(22) 出願日 平成10年(1998)3月16日

(31) 優先権主張番号 08/828449

(32) 優先日 1997年3月28日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72) 発明者 ジョーゼフ・エム・クライトン

アメリカ合衆国12603 ニューヨーク州ポークープシー エドガー・ストリート 9

(74) 代理人 弁理士 坂口 博 (外1名)

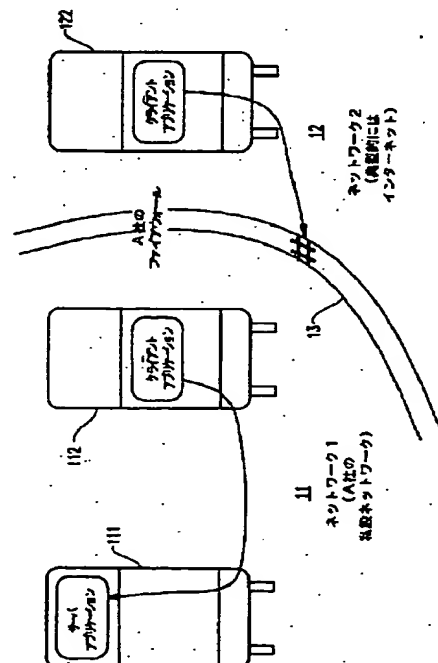
最終頁に続く

(54) 【発明の名称】 セキュリティ保護通信トンネリングの方法及び装置

(57) 【要約】

【課題】 軽セキュリティ保護トンネリング・プロトコル (LSTP) によって、中間サーバまたはプロキシを使用して1つまたは複数のファイアウォールを通る通信を可能にする。

【解決手段】 ファイアウォールを通してナビゲートする終端間接続を、3つのプロキシを使用して確立する。典型的な構成では、第1のファイアウォールの背後にあるサーバと第2のファイアウォールの背後にあるクライアントが、両ファイアウォール間の非信頼ネットワーク (たとえばインターネット) によって相互接続される。第1のファイアウォール内部 SOCKS 認識サーバ側終端プロキシが、第1のファイアウォールの内部にあるサーバに接続する。第2のファイアウォール内部 SOCKS 認識クライアント側終端プロキシが、第2のファイアウォールの内部にあるクライアントによって接続される。サーバ側とクライアント側の両方の終端プロキシが、2つのファイアウォールの外部にある第3のプロキシ (中間プロキシと呼ぶ) をアドレスすることができる。



## 【特許請求の範囲】

【請求項1】サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1のネットワークと第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む、第1のファイアウォールと、

相互にアドレス可能なサーバ終端プロキシ及びサーバ・アプリケーションと、

相互にアドレス可能なクライアント終端プロキシ及びクライアント・アプリケーションと、

第1のファイアウォールの外部にあり、第1のネットワークと第2のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含み、サーバ終端プロキシとクライアント終端プロキシがそれぞれ第1のファイアウォールを介して中間プロキシとの接続を行い、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバス・スルー通信トンネルを確立する、パケット交換ネットワーク通信システム。

【請求項2】第1のネットワークと第2のネットワークのうちの他方のネットワークのコンピュータ資源を保護し、第2のファイアウォールが第2のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む第2のファイアウォールをさらに含む、請求項1に記載のパケット交換ネットワーク通信システム。

【請求項3】サーバ終端プロキシと、クライアント終端プロキシと、中間プロキシがSOCKSサーバ機能を有するトンネルを構成し、トンネル全体がSOCKSサーバのジョブを実行する、請求項2に記載のパケット交換ネットワーク通信システム。

【請求項4】サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1と第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部に接続を行うことができるようにするソフトウェア・アプリケーションを含む第1のファイアウォールと、サーバ・アプリケーションによるアドレスが可能なサーバ終端プロキシと、クライアント・アプリケーションによるアドレスが可能なクライアント終端プロキシと、第1のファイアウォールの外部にあり、第1のネットワークと第2のネットワークの間の非信頼ネットワーク内にある中間プロキシとを

含むパケット交換ネットワーク通信システムにおいて、第1のファイアウォールを介して中間プロキシにサーバ終端プロキシとクライアント終端プロキシとを接続し、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバススルー通信トンネルを確立する方法であって、

中間プロキシを始動させ、終端プロキシからの第1の接続を待つステップと、

10 クライアント終端プロキシを始動させ、中間プロキシにクライアント・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、

中間プロキシによって、終端プロキシ・セットアップ情報を記憶し、その後で第2の接続を待つステップと、

サーバ終端プロキシを始動させ、中間プロキシに終端プロキシ・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、

20 中間プロキシによって、クライアント終端プロキシの接続とサーバ終端プロキシの接続とを対にし、サーバ及び中間プロキシ・セットアップ情報をクライアント終端プロキシに送信し、クライアント及び中間プロキシ・セットアップ情報をサーバ終端プロキシに送信するステップと、

その後で中間プロキシがクライアント終端プロキシとサーバ終端プロキシとの間のバス・スルーとして機能するステップとを含む方法。

【請求項5】中間プロキシによってクライアント終端プロキシとサーバ終端プロキシの間の接続を対にした後で、クライアント終端プロキシとサーバ終端プロキシによってセキュリティ・ハンドシェイクを交換するステップと、

セキュリティ保護された回線で中間プロキシを介してクライアント終端プロキシとサーバ終端プロキシの間でセットアップ情報を再び交換するステップとをさらに含む、請求項4に記載の方法。

【請求項6】トンネルを介したクライアントとサーバの間のデータ交換が完了したときにクライアント終端プロキシとサーバ終端プロキシとの間のトンネル資源を解放するステップをさらに含む、請求項5に記載の方法。

40 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般にはパケット交換ネットワーク通信に関し、詳細には、外部クライアントが内部サーバにアドレスすることができないように組織の「ファイアウォール」が構成されている場合であっても、組織の「ファイアウォール」の外部にあるTCP/IPクライアントがその同じファイアウォールの内部にあるサーバをアドレスすることができるようにする方法及び装置に関する。

50 【0002】

【従来の技術】インターネットは、政府機関、教育機関、及び民間企業を含む参加組織の間での資源の共有を容易にする世界中のネットワークの集まりである。これらのネットワークは、転送制御プロトコル／インターネット・プロトコル(TCP/IP)プロトコル群を使用し、共通アドレス空間を共用する。したがって、インターネット上のコンピュータは互換性のある通信標準を使用し、互いに連絡してデータを交換する能力を共用する。インターネットのユーザは主に、電子メール(eメール)や、ユーザがリモート・ホストにログ・インすることができるようになるプロセスであるTelnetや、リモート・ホスト上の情報をローカル・サイトに転送することができるようになるプロトコルであるファイル転送プロトコル(FTP)の実現を介して通信する。

【0003】ローカル・エリア・ネットワーク(LAN)などのネットワークをインターネットに接続する場合、セキュリティが大きな問題である。重要な問題の1つはローカル・ホストへのアクセスを獲得しようと試みる侵入者である。この種の侵入を防止する一般的な方法は、インターネットへのセキュリティ保護された単一の接続点である、いわゆる「ファイアウォール」を設置することである。この単一接続点は、ファイアウォール管理者によって指定された特定の通信のみを通過させるファイアウォール・ホストの形態をとる。典型的なファイアウォール・ホスト実施態様では、LAN内のホスト上のファイルをインターネットを介して外部ホストに転送しようとするユーザは、まずそのファイルをファイアウォール・ホストに転送し、次にファイアウォールにログ・インタフェースしてファイルを外部ホストに転送する。この手続きは単一ユーザに対しては高度なセキュリティを与えるが、そのホストへのアクセスを必要とするユーザの数が増えるにつれてセキュリティを維持するのが難しくなる。ファイアウォールの概要については、ウィリアムR. チェスウィック(William R. Cheswick)及びスティーブン・M. ベロヴィン(Steven M. Bellovin)著の「Firewalls and Internet Security」(Addison-Wesley, 1994)等に記載されている。

【0004】セキュリティの問題を最小限にすると同時に多数のユーザによるアクセスを可能にすることを目指して、SOCKSと呼ばれるトランスポート層プロキシ・アーキテクチャが作成された。これについては、たとえばデイヴィッド・コブラス(David Coblas)及びミシェルR. コブラス(Michelle R. Koblas)の「SOCKS」(UNIX Security Symposium, USENIX Association, 1992)、77~83ページ、ならびにインダ・リー(Ying-Da Lee)の「SOCKS: A protocol for TCP pr

oxy across firewalls」(http://www. socks. nec. com/socks4. protocol)、ならびにM. リーチ(Leech)、M. ガニス(Ganis)、Y. リー(Lee)、R. クリス(Kuris)、D. コブラス(Koblas)、及びL. ジョーンズ(Jones)の「SOCKS Protocol Version 5」(ftp://ds. internic. net/rfc/c/rfc1928. txt)等に記載されている。。トランスポート層プロキシ・アーキテクチャでは、ファイアウォールの背後にあるクライアントとよばれる1つの終端システムが、ファイアウォール上にあるとみなせるプロキシへの接続を行うことによってセッションを開始する。クライアントとプロキシはこの接続を使用してメッセージを交換し、認証やプロキシ要求などのセッション・セットアップ情報(たとえばファイアウォール・プロキシの場合には接続する外部ホスト、HTTP(ハイパーテキスト転送プロトコル)プロキシの場合にはフェッチするURL(ユニフォーム・リソース・ロケータ)など)について折衝する。次に、プロキシは、通常は、クライアントの指示に従って、サーバと呼ばれる一般にはファイアウォールの外部にある別の終端システムへの接続を開く要求を行う。プロキシは、この接続を介してサーバとセッション・セットアップ情報を交換することができる。両方の接続でセッション・セットアップが完了した後、プロキシはこの2つの接続間で相互にデータのコピーを開始し、ホスト間を流れる情報の削除、追加、または変更は行わない(ただし、HTTPキャッシュ・プロキシの場合のように情報のコピーを密かに保持することはできる)。

【0005】組織内の従業員は、「外部」クライアントが自分の「内部」サーバをアドレスすることができるようにしたいと考えることが多い。この場合、従業員は外部クライアントを信頼しているため、ファイアウォール上に設置された制御を迂回して、信頼された外部クライアントが信頼された内部サーバをアドレスすることができるようにしたいと考えることがある。

【0006】

【発明が解決しようとする課題】したがって、本発明の目的は、既存のファイアウォール・ソフトウェアやファイアウォール構成に変更を加えることなく、ファイアウォールを通過するセキュリティ保護されたトンネルを確立する方法を提供することである。

【0007】本発明の他の目的は、チャネルを介した通信の終端間プライバシー及び保全性のための備えを提供することである。

【0008】本発明の他の目的は、セキュリティ保護されたチャネルを確立するユーザの相互認証のための備えを提供することである。

【0009】本発明の他の目的は、ユーザIDとパスワ

10

20

30

40

50

ードをリモート・サーバに私的送達することができるようにするログイン／パスワードに基づくプロトコルを提供することである。

【0010】

【課題を解決するための手段】本発明によると、中間サーバまたはプロキシを使用して1つまたは複数のファイアウォールを通る通信を可能にする、「軽セキュリティ保護トンネリング・プロトコル」LSTP (Light weight Secure Tunneling Protocol) が提供される。具体的には、この基本システムは1つの中間プロキシ (middle proxy) と2つの終端プロキシ (end proxy) から成る3つのプロキシを使用して、2つのファイアウォールを通過してナビゲートする終端間接続を確立する。この構成では、第1のファイアウォールの背後にあるサーバと第2のファイアウォールの背後にあるクライアントが2つのファイアウォール間にある信頼性のないネットワーク (たとえばインターネット) によって相互接続される。第1のファイアウォール内部SOCKS認識終端サーバ側プロキシが、第1のファイアウォール内部サーバに接続する。第2のファイアウォール内部のクライアントは、第2のファイアウォール内部SOCKS認識クライアント側終端プロキシに接続する。サーバ側とクライアント側の両方の終端プロキシは、この2つのファイアウォールの外部にある第3のプロキシ (中間プロキシと呼ぶ) をアドレスすることができる。他の2つのプロキシ (サーバとクライアントの終端プロキシ) が始動後しばらくしてから中間プロキシへの接続を開始するため、通常は中間プロキシが先に始動される。中間プロキシは、両方の内部終端プロキシによって相互にアドレス可能であるため、サーバとクライアントの間に完全な終端間接続が確立される。1つまたは複数の中間プロキシをLSTPのような適切なプロトコルと共に使用することで、複数のファイアウォールを通るセキュリティ保護された通信リンクまたはトンネルが確立される。

【0011】

【発明の実施の形態】図面、特に図1を参照すると、ファイアウォール13によって分離された2つのネットワーク11及び12が図示されている。ファイアウォール13は、第1のネットワーク11をファイアウォールの外部から起こされる不正活動から保護するために使用される。この図では、ネットワーク11はA社の私設ネットワークであり、ネットワーク12は典型的にはインターネットである。この例では、ネットワーク11はサーバ・アプリケーションを稼働させているサーバ111と、クライアント・アプリケーションを稼働させているクライアント112によって表されている。ファイアウォール13の背後にあるクライアント112上で稼働しているクライアント・アプリケーションは、サーバ111上で稼働しているサーバ・アプリケーションをアドレ

スすることができるが、ファイアウォール13の外部にあるクライアント112上で稼働しているクライアント・アプリケーションはサーバ111上で稼働しているサーバ・アプリケーションをアドレスすることができない。これは、この通信がファイアウォールによって遮断されているためである。すなわち、ファイアウォール13の目的は、この場合はローカル・エリア・ネットワーク (LAN) 上で互いに接続されている多くのサーバとクライアントを含むことができるネットワーク11であるA社のコンピュータ資源を保護することである。

【0012】図2に、ネットワーク2内でサーバ・アプリケーションを稼働させているサーバ121が存在する点以外は同様の配置に図示されている。図1と同様に、クライアント112上で稼働しているクライアント・アプリケーションはこの場合も、サーバ111上で稼働しているサーバ・アプリケーションをアドレスすることができる。内部から外部への通信ができるように使用可能になっていれば、ファイアウォール13はクライアント112上で稼働しているクライアント・アプリケーションがファイアウォール13の外部にあるサーバ121上で稼働しているサーバ・アプリケーションに接続することができるようにもする。前述のSOCKSと呼ばれる共通ソフトウェア・パッケージによって、ファイアウォールは図2に示すように内部から外部への接続を行うことができる。

【0013】図3にも2つのネットワークが示されているが、第2のネットワーク14は今度は第2のファイアウォール15の背後にあるB社の私設ネットワークである。したがって、ファイアウォール15はB社のコンピュータ資源をファイアウォール15の外部から起こされる不正な活動から保護するように設計されている。この図では、ファイアウォール15の背後にあるクライアント142上で稼働しているクライアント・アプリケーションは、インターネット12を介してファイアウォール13の背後にあるサーバ111上で稼働しているサーバ・アプリケーションへのアドレスを試みることができる。しかし、ファイアウォール15はクライアント142上で稼働しているクライアント・アプリケーションがファイアウォール15の外部に接続することができるようにするSOCKS機能を備えることができるのに対し、ファイアウォール13はサーバ111上で稼働しているサーバ・アプリケーションへの接続を禁止する。

【0014】組織内部の従業員は「外部」クライアント・アプリケーションが「内部」サーバをアドレスすることができるようにしたいと考えることが多い。たとえば、図3のクライアント142上で稼働しているクライアント・アプリケーションがサーバ111上で稼働しているサーバ・アプリケーションをアドレスすることができるようにしたい場合がある。この場合、従業員は外部クライアント・アプリケーションを信頼し、信頼されて

いる外部クライアントが内部サーバをアドレスするのを禁止する自社のファイアウォールに設置された制御を迂回したいと考える。本発明は、このような状況の解決策を提供する。

【0015】以下に、本発明について一般性を失うことなく、X Windows Systemにおける実施態様に関して説明する。X Windows Systemは、UNIXワークステーション用にMITで開発され、ハードウェア独立グラフィカル・ユーザ・インタフェース（GUI）の作成を可能にする表示処理ルーチンの標準セットである。最初に説明する例では、2つのファイアウォールを扱う。この最初の実施態様は、SOCKSゲートウェイ内部のSOCKS認識プログラムがSOCKSゲートウェイ外部のサーバに接続することができるようにするSOCKSパッケージを基に構築されている。

【0016】図4に、本発明による、セキュリティ保護されたトンネルを構築するために使用する3つのタイプのプロキシとネットワーク構成におけるその配置とを示す。A社の私設ネットワークまたはイントラネットであるネットワーク21はファイアウォール23によって保護され、B社の私設ネットワークまたはイントラネットであるネットワーク22はファイアウォール25によって保護されている。ファイアウォール23の背後にはA社の私設ネットワークの一部であるXサーバ211があり、ファイアウォール25の背後にはB社の私設ネットワークの一部であるXクライアント222がある。ファイアウォール23と25は両方とも、「内部」クライアントが「外部」サーバに接続することができるようにするSOCKS機能を備えている。Xクライアント222はB社のファイアウォール25の背後にあるそのアドレス可能ドメイン内に、Xプロトコルの着信を監視することができるクライアント終端プロキシ223を持っている。クライアント終端プロキシ223は、Xクライアント222にとってローカルXサーバであるように見えるため、Xクライアント222には修正を加える必要がない。

【0017】Xサーバ211のアドレス可能ドメイン内にサーバ終端プロキシ213があるA社のファイアウォール23の背後にも同様の状況が存在する。サーバ終端プロキシ213は、実Xクライアントとまったく同様にXサーバ211に接続することができる。サーバ終端プロキシ213はXサーバ211にとってローカルXクライアントのように見えるため、この場合もXサーバ211には修正を加える必要がない。

【0018】終端プロキシが中間プロキシへの接続を開始することになるため、中間プロキシ26が先に始動する。クライアント終端プロキシ223とサーバ終端プロキシ213は既存の機能（たとえばSOCKS）を使用して内部から外部にファイアウォールを通して要求を行

う。中間プロキシ26は（各ファイアウォール上のSOCKSを使用して）両方の終端プロキシによって相互にアドレス可能であるため、中間プロキシ26を介してXクライアント222とXサーバ211の間に完全な終端間接続を確立することができる。

【0019】中間プロキシ26はクライアント終端プロキシ223とサーバ終端プロキシ213の両方にとってサーバのように見え、本発明の重要な特徴である。したがって、クライアント終端プロキシ223とサーバ終端プロキシ213は両方ともそれぞれのファイアウォール25及び23を通過して中間プロキシ26をアドレスすることができる。カスケード中間プロキシまたは複数の中間プロキシの場合、中間プロキシは終端プロキシによってアドレスされるのではなく、実際に他の中間プロキシをアドレスする。標準TCP/IP接続機構を使用して初期接続が行われる。確立された各接続は、どのプログラムによって開始されたかを問わず、TCP/IP接続であり、したがって全二重である。本発明は、TCP/IP上で使用してトンネル管理事象の適切なセキュリティ保護を行う「軽セキュリティ保護トンネリング・プロトコル」（LSTP）を提供する。LSTPは、トンネル構築中だけでなくトンネル存続期間全体を通じて、クライアント終端プロキシと、サーバ終端プロキシと、中間プロキシの間で「使われる」。

【0020】終端プロキシ213と223が中間プロキシ26への接続を開始するトリガは、終端プロキシ213及び223が稼働しているコンピュータへのアクセス権を持っている人によって主導制御される。終端プロキシ213及び223は、中間プロキシが始動された後いつでも中間プロキシ26との接続を確立することができる。中間プロキシ26は最初に接続する終端プロキシから送られたセットアップ情報を受信し、記憶する。

【0021】中間プロキシが、クライアント終端プロキシ223からの接続とサーバ終端プロキシ213からの接続の2つの一致する接続を持っている場合、中間プロキシ26はその2つの接続に加わり、透明なパイプのような役割を果たして、実質的に2つの終端プロキシの間に接続を確立する。この時点以降、中間プロキシ26はパス・スルー・モードになり、一方の終端プロキシが他方の終端プロキシとのセキュリティ・ハンドシェイクを開始してこのトンネルをセキュリティ保護する。

【0022】これでXクライアント222はクライアント終端プロキシ223と接続することができ、Xサーバ211に直接接続されているかのように終端プロキシ223にデータを渡す。Xクライアント222からの初期データによって、2つの終端プロキシ223及び213と中間プロキシ26によって確立されたトンネルを通してLSTPメッセージが流れ、次にそれによってサーバ終端プロキシ213がXサーバ211との接続を開始する。その後、Xクライアント222からのデータがトン

ネルを通過し、Xサーバ211に渡される。互いに直接通信していないことを示すものは、Xクライアント222にもXサーバ211にもまったくない。データは、Xサーバ211からXクライアント222へと、Xクライアント222からXサーバ211への両方向に流れる。この時点で、追加のクライアントがクライアント・プロキシに接続して同じトンネル接続を使用したり、新しいトンネル接続を要求したりすることができる。

【0023】要約すると、サーバ側終端プロキシは内部X Windows Systemサーバ及び外部中間プロキシと接続することができ、X Windows Systemクライアントはクライアント側終端プロキシと接続することができ、さらにクライアント側終端プロキシがXクライアントのために外部中間プロキシと接続することができる。確立される接続が本質的に全二重であるためと、過渡的な閉鎖状態のため、X Windows SystemクライアントはファイアウォールがないかのようにX Windows Systemサーバをアドレスすることができる（すなわち、同じアドレス可能ネットワーク上にあるかのようなものである）。当業者なら、他のプロトコルやサービスに対応できるように終端プロキシの機能を増強することができるであろう。たとえば、1つの終端プロキシがクライアントとサーバの両方の終端プロキシ機能を備えることもできる。

【0024】図5は、クライアント・プロキシと中間プロキシとサーバ・プロキシとの間の対話を示すデータ流れ図である。このプロセスは、中間プロキシが始動されており、最初の接続を待っている状態を仮定している。クライアント終端プロキシが始動され、中間プロキシとの接続を開き、中間プロキシにクライアント・セットアップ情報を送る。「セットアップ情報」とは、2つのLSTPプロトコルを説明する一般用語である。中間プロキシは終端プロキシ・セットアップ情報を記憶した後、2番目の接続を待つ。サーバ終端プロキシが始動され、中間プロキシとの接続を開き、サーバ終端プロキシが中間プロキシに終端プロキシ・セットアップ情報を送る。中間プロキシはこの2つの接続を対にし、クライアント終端プロキシにサーバ・プロキシと中間プロキシのセットアップ情報を送信し、サーバ終端プロキシにクライアント・プロキシと中間プロキシのセットアップ情報を送信する。

【0025】プロセスのこの時点で、中間プロキシは接続における積極的な役割を終え、クライアント終端プロキシとサーバ終端プロキシとの間のパス・スルーの役割を果たす。接続が確立された後は、いずれか一方の終端プロキシからセキュリティ・ハンドシェイクを開始することができる。各終端プロキシ内のアルゴリズムがセットアップ情報を使用してどちらの終端プロキシがセキュリティ・ハンドシェイクを開始するかを決定する。図5

には、クライアント終端プロキシがセキュリティ・ハンドシェイクを送信し、それがサーバ終端プロキシに渡されるように図示されている。サーバ終端プロキシは、セキュリティ・ハンドシェイクで応答し、それがクライアント終端プロキシに渡される。このセキュリティ・プロトコルが遂行されると、セキュリティ保護された回線を介して追加のセットアップ情報が再送信され、トンネル構築が完了する。当業者なら、トンネルを介した終端間セキュリティを確立するために他の事象シーケンスも使用可能であることがわかるであろう。クライアント・アプリケーションとサーバ・アプリケーションとの接続が完了すると、後述の「軽セキュリティ保護トンネリング・プロトコル」(LSTP)などのプロトコルを使用して2つのアプリケーションの間で双方向にデータを安全に受け渡しすることができる。その結果、クライアント・アプリケーションとサーバ・アプリケーションはそれぞれのアドレス可能性を実質的に拡張させたことになる。

【0026】サーバ・プロキシ213とクライアント・プロキシ223は、(1)認証、暗号化、及び保全と、(2)ファイアウォールの通過と、(3)データ圧縮とを扱う。中間プロキシ26は双方向パイプとして機能する。サーバ終端プロキシ213はXサーバ211にとってXクライアントのように見え、クライアント終端プロキシ223はXクライアント222にとってXサーバのように見える。サーバ終端プロキシ及びクライアント終端プロキシは通常、それぞれXサーバ及びXクライアントと同じマシン上にある。

【0027】この概説では、2つのネットワーク21と22の間にセキュリティ保護トンネルを実現するためにいくつかの実施態様上の決定を行った。第1に、この例では、トンネル両端でX Windows Systemクライアントとサーバを使用した。したがって、終端プロキシはXプロトコルの着信を監視し、それに応答するようにカスタマイズされている。第2に、トンネルをセキュリティ保護するセキュリティ・プロトコルとしてセキュア・ソケット層(SSL)を選定した。SSLは、発信側のデータ保全、データ・プライバシー、及び認証を行う。第3に、プロキシがファイアウォールの内部からファイアウォールの外部への接続を確立することができるようにする機構としてSOCKSを選定した。第4に、トンネル構築、管理、データ・フロー制御、及びトンネル破棄を定式化する手段を提供するために、トンネルと共に「軽セキュリティ保護トンネリング・プロトコル」(LSTP)を開発した。

【0028】本発明による「軽セキュリティ保護トンネリング・プロトコル」(LSTP)は、図4に示すように、クライアント・プロキシ223と中間プロキシ26との間と、サーバ・プロキシ213と中間プロキシ26との間で使用するプロトコルである。好ましい実施例で

は、LSTPはプロキシ間のデータ転送とプロキシの状態の同期化のために、要求と応答について以下のような意味と順序づけ規則を含む。

COMMONUNIQUE\_INFO - この情報によって2つの終端プロキシが同じ中間プロキシに異なる時点で接続することができる。両方のプロキシが同じ共通情報を提供したため、中間プロキシはこの2つの終端プロキシを互いに対にすべきであることを認識する。固有情報を使用して各終端プロキシ・ユーザを識別することもできる。

TOPOLOGY\_EXCHANGE - これはトンネルのトポロジを記述する情報である。中間プロキシは受け取ったどのTOPOLOGY\_EXCHANGEにもそのトポロジ（たとえば名前、アドレスなど）情報を付加し、そのTOPOLOGY\_EXCHANGEを終端プロキシまで転送する。これによって、各終端プロキシに、どのプロキシがトンネルに関与しているかを示すマップが提供される。

PROPERTY\_EXCHANGE - これは、終端プロキシがそれ自体に関する情報を交換することができるようにする機構である。

CONNECTION\_REQUEST - この要求によって一方の終端プロキシが他方の終端プロキシに対して、クライアント・アプリケーションがクライアント・アプリケーションによって使用されるトンネル資源の割り振りを要求していることを通知することができる。要求される資源には、既存のトンネル接続上の「多重化」チャンネルまたは確立済みトンネル接続に追加される新規トンネル接続が含まれる。

CONNECTION\_ACK及びCONNECTION\_NACK - この2つの応答によって、CONNECTION\_REQUESTを受け取るプロキシはトンネル資源を求める要求を受け入れまたは拒否することができる。

SERVICE\_BEGIN\_REQUEST - この要求によって、終端プロキシは、アプリケーションがデータの送信を開始し、したがって要求または割り振りされたトンネル資源の使用を開始することを他方の終端プロキシに通知することができる。

SERVICE\_BEGIN\_ACK及びSERVICE\_BEGIN\_NACK - この2つの応答によって、終端プロキシは割り振られている資源の使用を開始するアプリケーションの要求を受け入れまたは拒否することができる。

SERVICE\_DATA - このメッセージは、2つの終端プロキシ間でアプリケーション・データを送信するために使用される。各クライアント/サーバ・アプリケーションの対がその対のSERVICE\_DATAメッセージ内に固有の識別子を含んでおり、それによって複数のアプリケーションが1つのTCP/IP接続を

介してそれらのデータを多重化することができる。

SERVICE\_DATA\_PAUSE - このメッセージによって、終端プロキシは他方の終端プロキシにアプリケーション・データの送信を停止するように指示することができる。

SERVICE\_DATA\_RESUME - このメッセージによって終端プロキシは他方の終端プロキシにアプリケーション・データの送信を再開するように指示することができる。

10 SERVICE\_FREE\_REQUEST - この要求によって、終端プロキシは他方の終端プロキシに、アプリケーションが完了し、トンネル資源を解放することができることを通知することができる。

SHUTDOWN - これによって、終端プロキシは他方の終端プロキシに通知することによってトンネルを正常に遮断することができる。

ERROR - このメッセージによって、終端プロキシはエラー情報を交換することができる。

20 【0029】LSTPを図6にまとめる。最初の3つのメッセージはトンネルのセットアップと管理を扱うために使用されることに留意されたい。具体的には、セットアップ情報は、COMMONUNIQUE\_INFOとTOPOLOGY\_EXCHANGE LSTPメッセージの様々な組み合わせからなり、トンネル接続を扱うために使用される。次の6つのメッセージは、トンネルを使用するアプリケーションの必要に応じて既存のトンネルの資源を扱うために使用される。その次のメッセージSERVICE\_DATAは、終端プロキシ間でアプリケーション・データを伝送するために使用される。その次の2つのメッセージは、アプリケーション・データの管理、すなわちフロー制御に使用される。次の3つのメッセージは、アプリケーションが必要としなくなったトンネル資源のクリーンアップを扱うために使用される。最後に、最終メッセージはエラー状態を管理するために使用される。

【0030】この「軽セキュリティ保護トンネリング・プロトコル」（LSTP）は、トンネル資源管理とライフ・サイクルを支援するために開発された。当業者なら、同じ目的を達成するために同様の特徴と機能を含む他のプロトコルを作成することもできることがわかるであろう。

【0031】終端プロキシの流れ図を図7に示す。このプロセスは機能ブロック701で中間プロキシに接続することによって開始される。次に、機能ブロック702で終端プロキシは自分のCOMMONUNIQUE\_INFO及びTOPOLOGY\_EXCHANGEセットアップ情報を中間プロキシに送る。次に、機能ブロック703で終端プロキシは他のプロキシのCOMMONUNIQUE\_INFO及びTOPOLOGY\_EXCHANGEセットアップ情報を中間プロキシから受け取



る。機能ブロック704で、TOPOLOGY\_EXCHANGEセッアップ情報に基づいてマスタ終端プロキシが選定される。判断ブロック705でこのプロキシがマスタ終端プロキシであると判断された場合、機能ブロック706でセキュリティ・ハンドシェークが開始される。そうでない場合には、機能ブロック707でこの終端プロキシはセキュリティ・ハンドシェークを待つ。機能ブロック708でセキュリティ・ハンドシェークが完了すると、機能ブロック709で終端プロキシはセキュリティ保護された接続を介してCOMMONUNIQUE\_10 INFO及びTOPOLOGY\_EXCHANGEセッアップ情報を再送信する。次に、機能ブロック710でプロキシはセキュリティ保護された接続を介して再びCOMMONUNIQUE\_10 INFO及びTOPOLOGY\_EXCHANGEセッアップ情報を受け取る。判断ブロック711での判断によりこの終端プロキシがクライアント終端プロキシである場合、機能ブロック712でこのプロキシはローカル・クライアント・アプリケーションが接続するのを待つ。機能ブロック713でローカル・クライアント・アプリケーションが接  
続されると、機能ブロック714で「軽セキュリティ保護トンネル・プロトコル(LSTP)」を使用してその接続がセッアップされ、管理される。それに対して、判断ブロック711での判断によりこのプロキシがサーバ終端プロキシである場合、機能ブロック715でプロキシは他方の終端プロキシが接続を要求するのを待つ。機能ブロック716で他方の終端プロキシからCONNECTION\_REQUESTメッセージを受け取ると、  
「軽セキュリティ保護トンネル・プロトコル(LSTP)」を使用してその接続がセッアップされ、管理される。

【0032】中間プロキシの流れ図を図8に示す。このプロセスは、判断ブロック801で近隣プロキシからの新しい接続がないかどうかを検査することによって始まる。新しい接続がある場合、機能ブロック802でその新しい接続が受け入れられ、判断ブロック803で、前の接続からの一致するCOMMONUNIQUE\_10 INFOメッセージが記憶されているかどうかを判断する。記憶されていない場合、機能ブロック804でCOMMONUNIQUE\_10 INFO及びTOPOLOGY\_EXCHANGEメッセージが中間プロキシに記憶され、プロセスは判断ブロック801に戻り、新しい接続を待つ。新しい接続を受信したとき、判断ブロック803での判断によって受信したCOMMONUNIQUE\_10 INFOが前に記憶されていたCOMMONUNIQUE\_10 INFOと一致する場合、機能ブロック805でその2つの接続を対にして2つの近隣プロキシ間に経路を確立する。一致したCOMMONUNIQUE\_10 INFOは機能ブロック806で記憶域から除去される。機能ブロック807で、COMMONUNIQUE\_10 INFO

及びTOPOLOGY\_EXCHANGEセッアップ情報が2つの新たに一致した近隣プロキシ間で交換され、判断ブロック801に戻る。ここで、近隣プロキシからの新しい接続ではないメッセージを受信した場合、判断ブロック808で、着信データを持つ既存の接続対が処理を待っていないかどうかを調べる検査を行う。そのような接続対がない場合は判断ブロック801に戻る。ある場合は、機能ブロック809で中間プロキシはパス・スルーとして機能する。すなわち、中間プロキシは一方の接続から着信データを受け取り、そのデータを接続された終端プロキシの対の他方の接続で送出する。

【0033】説明を簡単にするために1つの中間プロキシを示した。しかし、複数の中間プロキシをカスケードして、複数のファイアウォールを通る単一のトンネルを構築することもできる。この場合、それらの中間プロキシのうちのいくつかを、第2の着信接続を待つのではなく近隣中間プロキシとの接続を確立するように構成する。代替実施例では、1つのファイアウォールをアドレスする。これを図9に示す。この場合、A社の私設ネットワークまたはイントラネット21はXサーバ211と、ファイアウォール23の背後にあるサーバ終端プロキシ213とを含む。Xクライアント27はXサーバ211上で稼働しているアプリケーションをアドレスしようとしている。結合クライアント終端兼中間プロキシ28を使用してこの接続を容易にする。同様に、図9はXクライアントとXサーバの役割が逆転している場合を考えることができる。

【0034】本発明のこの代替実施例による解決策では、2つのプログラムを使用する。一方のプログラムはX Windows Systemクライアントのプロキシとして機能し、他方のプログラムはX Windows Systemサーバのプロキシとして機能する。クライアント側プロキシは外部ネットワーク（すなわちインターネット）上で稼働し、したがってX Windows Systemクライアントによってアドレスされることができる。この外部クライアント側プロキシは、ファイアウォールを通して内部のサーバ側プロキシによってもアドレスされることができる。これは、ほとんどのファイアウォールがSOCKSパッケージを使用して、内部SOCKS認識プログラムが外部サービスをアドレスして接続するためのゲートウェイを提供しているためである。このプロセスは大体前述の通りである。

【0035】図10に示すように、さらに増強することによって、FTP、TELNET、HTTP、及びSMTPなどのインターネット・プログラムの追加のSOCKS認識版が、セキュリティ保護されたトンネルを通して1つまたは複数のファイアウォールをナビゲートできるようになる。この改良によって、SOCKSサーバの機能がトンネル全体に付加され、それによって既存のSOCKS認識TCP/IPクライアントはトンネルの他

端にあるサーバと通信することができ、トンネル自体が2つ以上のファイアウォールをナビゲートすることがある場合であっても、トンネル自体を単一のファイアウォールとして扱うことができる。中間プロキシ26がパススルーとして機能し始めた後、サーバ終端プロキシ223とクライアント終端プロキシ213との間でLSTPが使用される。この実施例では、中間プロキシ26と2つの終端プロキシ213及び223は、SOCKSサーバ261を構成する。2つの終端プロキシは、SOCKS認識クライアント・アプリケーションからの送信を監視し、そのアプリケーションと通信するようにカスタマイズする必要がある。

【0036】RTELNET、RFTP、RHTTP（接頭字「R」は通常クライアントがSOCK化されていることを示す）などのようにSOCK化された多くの既存のクライアントがある。これらのクライアントは、典型的にはファイアウォール上にあるSOCKSサーバとの接続を期待しており、したがってSOCKSサーバにSOCKSプロトコルを送信することによって開始し、SOCKSサーバが応答することを期待する。

【0037】トンネルにSOCKSサーバ機能を付加することによって、トンネル全体がSOCKSサーバのジョブを行うようになる。したがってユーザは、RTELNETを使用してファイアウォールに接続する代わりに、RTELNETを使用してSOCKS使用可能トンネルに接続し、そのトンネル・インスタンスが通過する数のファイアウォールを通過することができる。この手続きのもう一つの考え方は、トンネルの機能に1つのプロトコル（すなわちSOCKS）を付加し、TELNET、ETP、HTTPなどのプロトコル群全体を獲得することである。SOCKSがなければ、TELNET、FTP、HTTPなどを認識して応答するようにトンネルを修正する必要が生じることになる。さらに、非SOCKS認識クライアントがSOCKSサーバに対して要求を行うことができるようにする市販の製品を使用して、どのようなクライアント・アプリケーションでもSOCKS認識終端プロキシにアクセスできるようにすることができる。

【0038】以上、本発明についてUNIXワークステーション上で稼働する場合について説明した。すなわち、サーバ、クライアント、終端プロキシ、及び中間プロキシはすべてUNIXワークステーション上で実施されている。しかし、たとえば、特定の用途や使用可能な資源によってはプロキシがパーソナル・コンピュータ（PC）またはメイン・フレーム・コンピュータ上で稼働できない理由はない。これは、UNIX基盤から別のプラットフォームにコードを移植するだけのことである。一般に、各プロキシが異なるハードウェア/ソフトウェア・プラットフォーム上で稼働しており、終端プロキシ、中間プロキシ、及び他の終端プロキシがトンネル

を構成することができる。終端プロキシと中間プロキシは両方とも、それらが稼働しているコンピュータの資源を比較的要求しない。

【0039】まとめとして、本発明の構成に関して以下の事項を開示する。

【0040】（1）サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1のネットワークと第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む、第1のファイアウォールと、相互にアドレス可能なサーバ終端プロキシ及びサーバ・アプリケーションと、相互にアドレス可能なクライアント終端プロキシ及びクライアント・アプリケーションと、第1のファイアウォールの外部にあり、第1のネットワークと第2のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含み、サーバ終端プロキシとクライアント終端プロキシがそれぞれ第1のファイアウォールを介して中間プロキシとの接続を行い、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にパス・スルー通信トンネルを確立する、パケット交換ネットワーク通信システム。

（2）第1のネットワークと第2のネットワークのうちの他方のネットワークのコンピュータ資源を保護し、第2のファイアウォールが第2のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む第2のファイアウォールをさらに含む、上記（1）に記載のパケット交換ネットワーク通信システム。

（3）サーバ終端プロキシと、クライアント終端プロキシと、中間プロキシがSOCKSサーバ機能を有するトンネルを構成し、トンネル全体がSOCKSサーバのジョブを実行する、上記（2）に記載のパケット交換ネットワーク通信システム。

（4）サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1と第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部に接続を行うことができるようにするソフトウェア・アプリケーションを含む第1のファイアウォールと、サーバ・アプリケーションによるアドレスが可能なサーバ終端プロキシと、クライアント・アプリケーションによるアドレスが可能なクライアント終端プロキシと、第1のファイアウォールの外部

にあり、第1のネットワークと第2のネットワークの間の非信頼ネットワーク内にある中間プロキシを含むバケット交換ネットワーク通信システムにおいて、第1のファイアウォールを介して中間プロキシにサーバ終端プロキシとクライアント終端プロキシとを接続し、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバススルー通信トンネルを確立する方法であって、中間プロキシを始動させ、終端プロキシからの第1の接続を待つステップと、クライアント終端プロキシを始動させ、中間プロキシにクライアント・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、中間プロキシによって、終端プロキシ・セットアップ情報を記憶し、その後で第2の接続を待つステップと、サーバ終端プロキシを始動させ、中間プロキシに終端プロキシ・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、中間プロキシによって、クライアント終端プロキシの接続とサーバ終端プロキシの接続とを対にし、サーバ及び中間プロキシ・セットアップ情報をクライアント終端プロキシに送信し、クライアント及び中間プロキシ・セットアップ情報をサーバ終端プロキシに送信するステップと、その後で中間プロキシがクライアント終端プロキシとサーバ終端プロキシとの間のバス・スルーとして機能するステップとを含む方法。

(5) 中間プロキシによってクライアント終端プロキシとサーバ終端プロキシの間の接続を対にした後で、クライアント終端プロキシとサーバ終端プロキシによってセキュリティ・ハンドシェイクを交換するステップと、セキュリティ保護された回線で中間プロキシを介してクライアント終端プロキシとサーバ終端プロキシの間でセットアップ情報を再び交換するステップとをさらに含む、上記(4)に記載の方法。

(6) トンネルを介したクライアントとサーバの間のデータ交換が完了したときにクライアント終端プロキシとサーバ終端プロキシとの間のトンネル資源を解放するステップをさらに含む、上記(5)に記載の方法。

【図面の簡単な説明】

【図1】サーバがファイアウォールの背後にある場合の、クライアント・アプリケーションとサーバ・アプリ

ケーションとの間の典型的な対話を示すブロック図である。

【図2】クライアントがファイアウォールの背後にある場合の、クライアント・アプリケーションとサーバ・アプリケーションとの間の典型的な対話を示すブロック図である。

【図3】それぞれがファイアウォールの背後にあるネットワークを有する2つの会社または組織の間の典型的なネットワーク構成を示すブロック図である。

【図4】2つの会社または組織の間にセキュリティ保護された通信チャネルまたはトンネルを構築するために本発明により使用する3つのタイプのプロキシを示すブロック図である。

【図5】図4に示すクライアント・プロキシ、中間プロキシ、及びサーバ・プロキシ間の対話を示すデータ流れ図である。

【図6】本発明の好ましい実施例による、「軽セキュリティ保護トンネリング・プロトコル」(LSTP)の概要を示す表である。

【図7】終端プロキシ上で実行されるプロセスを示す流れ図である。

【図8】中間プロキシ上で実行されるプロセスを示す流れ図である。

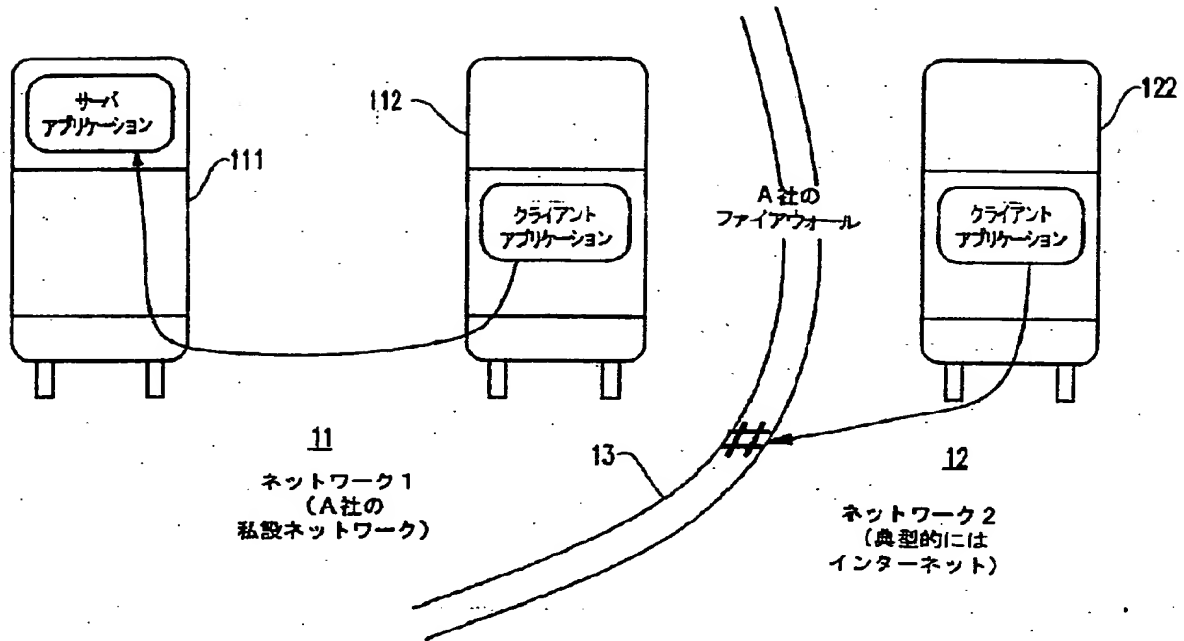
【図9】サーバ・プロキシが単一のファイアウォールを介してクライアント・プロキシに接続する本発明の代替実施例を示すブロック図である。

【図10】SOCKS認識クライアントが2つのファイアウォールを介したセキュリティ保護通信を可能にする本発明の他の代替実施例を示すブロック図である。

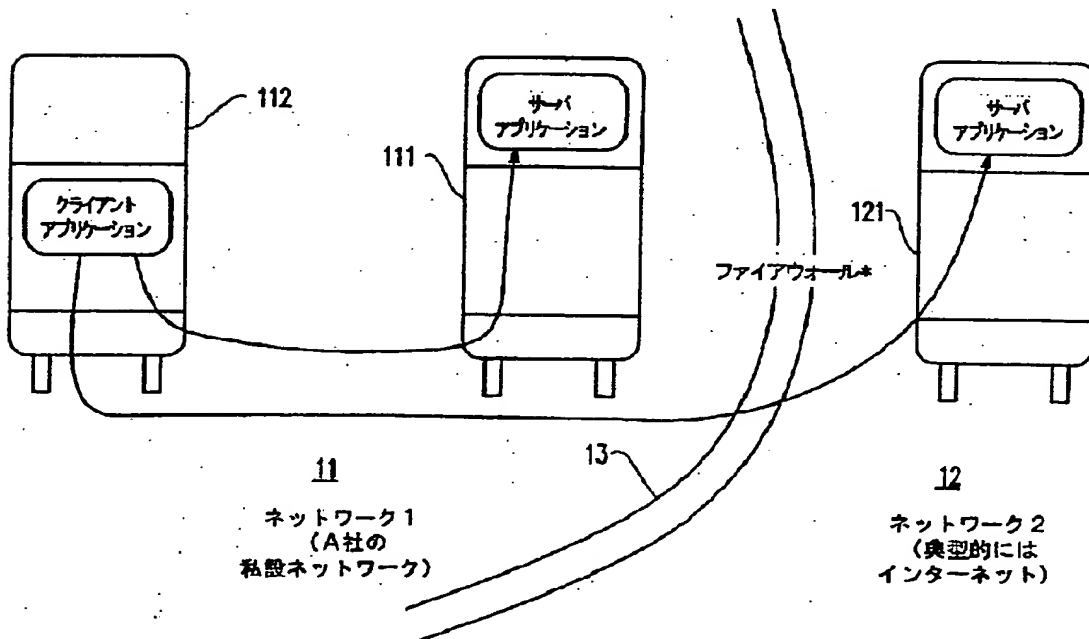
【符号の説明】

- 21 ネットワーク
- 22 ネットワーク
- 23 ファイアウォール
- 25 ファイアウォール
- 26 中間プロキシ
- 211 Xサーバ
- 213 サーバ終端プロキシ
- 222 Xクライアント
- 223 クライアント終端プロキシ

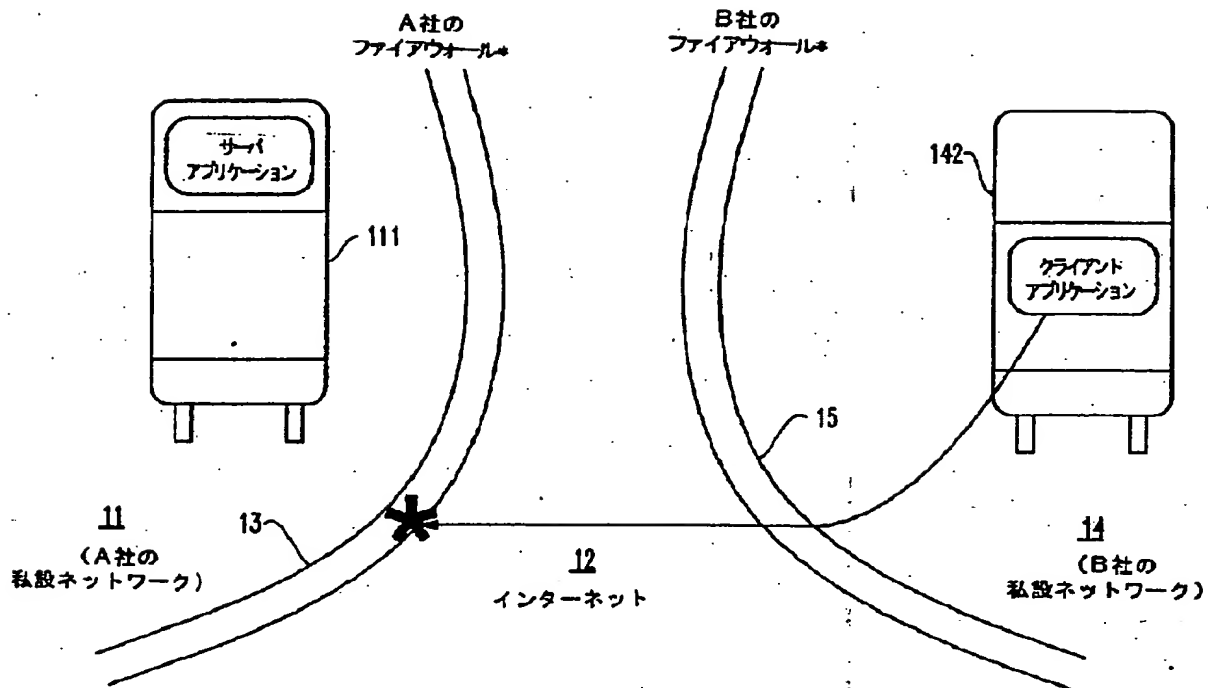
【図1】



【図2】



〔図3〕

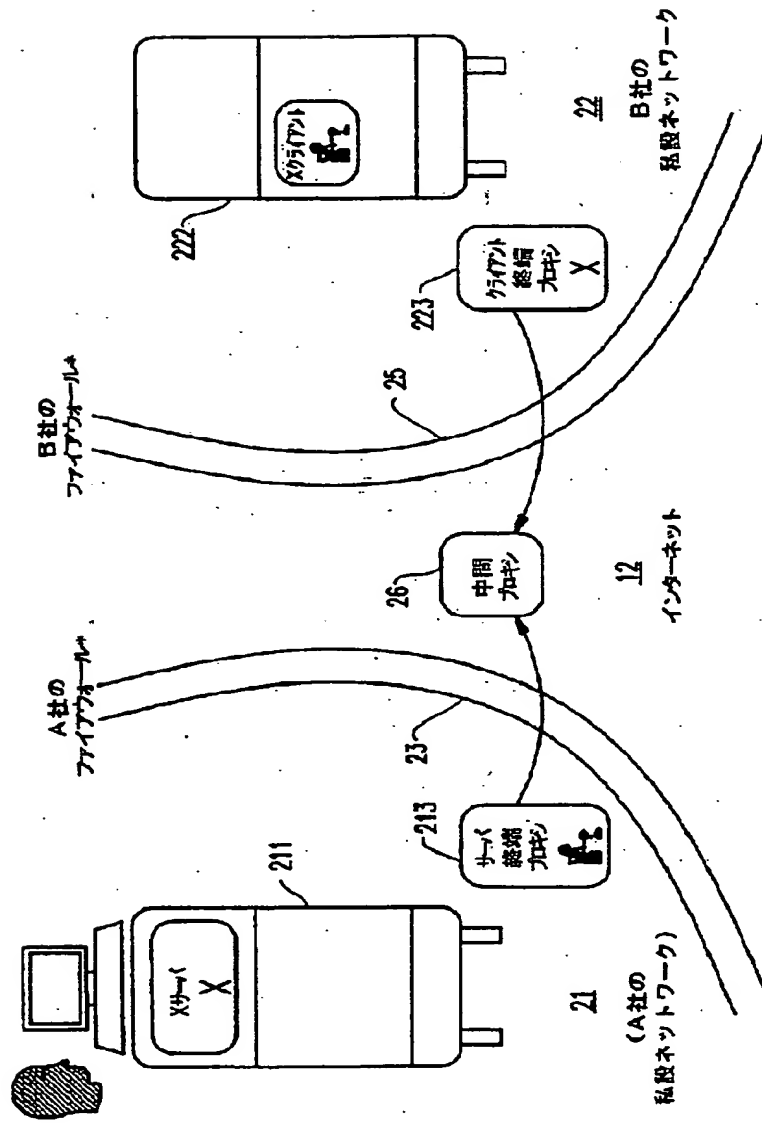


〔図6〕

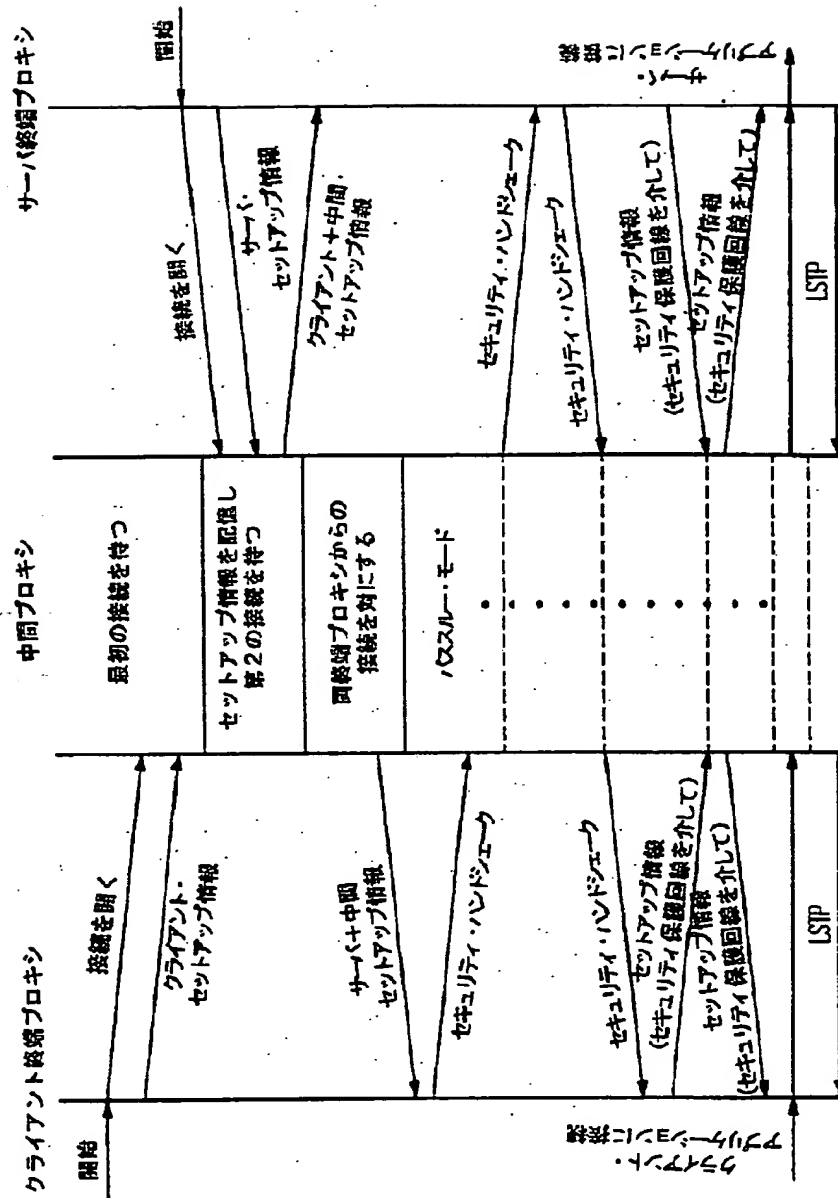
LSTPメッセージメッセージの一般的目的

|                       |   |   |
|-----------------------|---|---|
| COMMONUNIQUE_INFO     | — | トンネルセットアップと<br>管理を扱う                        |
| TOPOLOGY_EXCHANGE     |   |   |
| PROPERTY_EXCHANGE     |   |   |
| CONNECTION_REQUEST    | — | トンネルを使用するアプリケーションの<br>必要に応じて既存のトンネルの資源を管理する |
| CONNECTION_ACK        |   |   |
| CONNECTION_NACK       |   |   |
| SERVICE_BEGIN_REQUEST |   |   |
| SERVICE_BEGIN_ACK     |   |   |
| SERVICE_BEGIN_NACK    | — | 終端プロキシ間でアプリケーション<br>・データを伝送する               |
| SERVICE_DATA          |   |   |
| SERVICE_DATA_PAUSE    | — | アプリケーション・データを管理する<br>— フロー制御                |
| SERVICE_DATA_RESUME   |   |   |
| SERVICE_FREE_REQUEST  | — | アプリケーションが必要としなくなった<br>トンネル資源のクリーンアップを管理する   |
| SERVICE_FREE_RESPONSE |   |   |
| SHUTDOWN              |   |   |
| ERROR                 | — | エラー条件を管理する                                  |

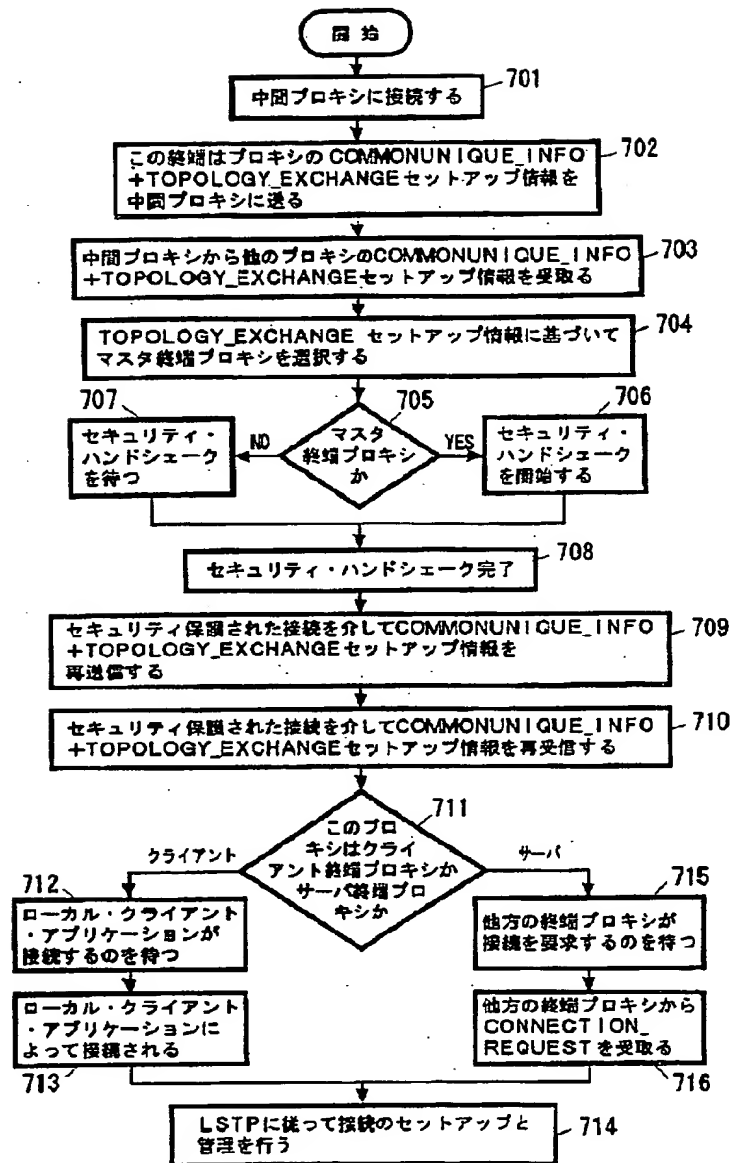
〔図4〕



【図5】

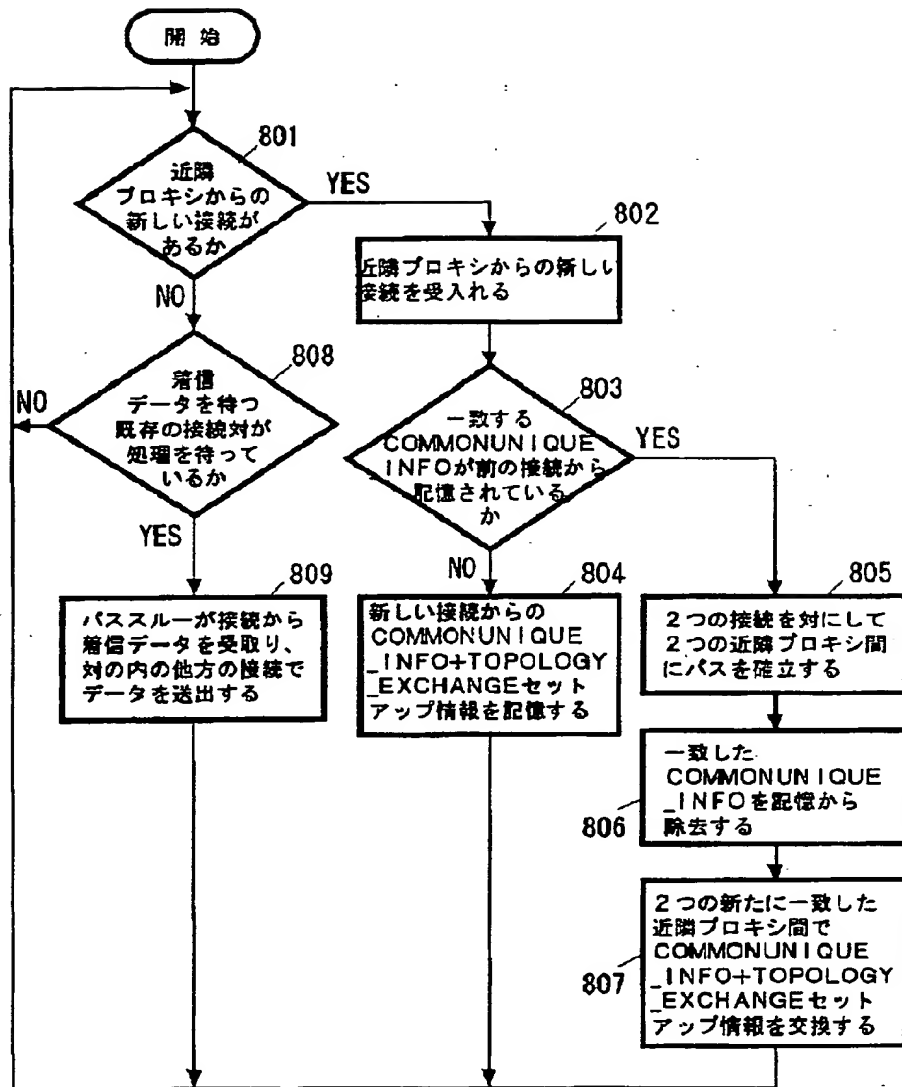


【図7】

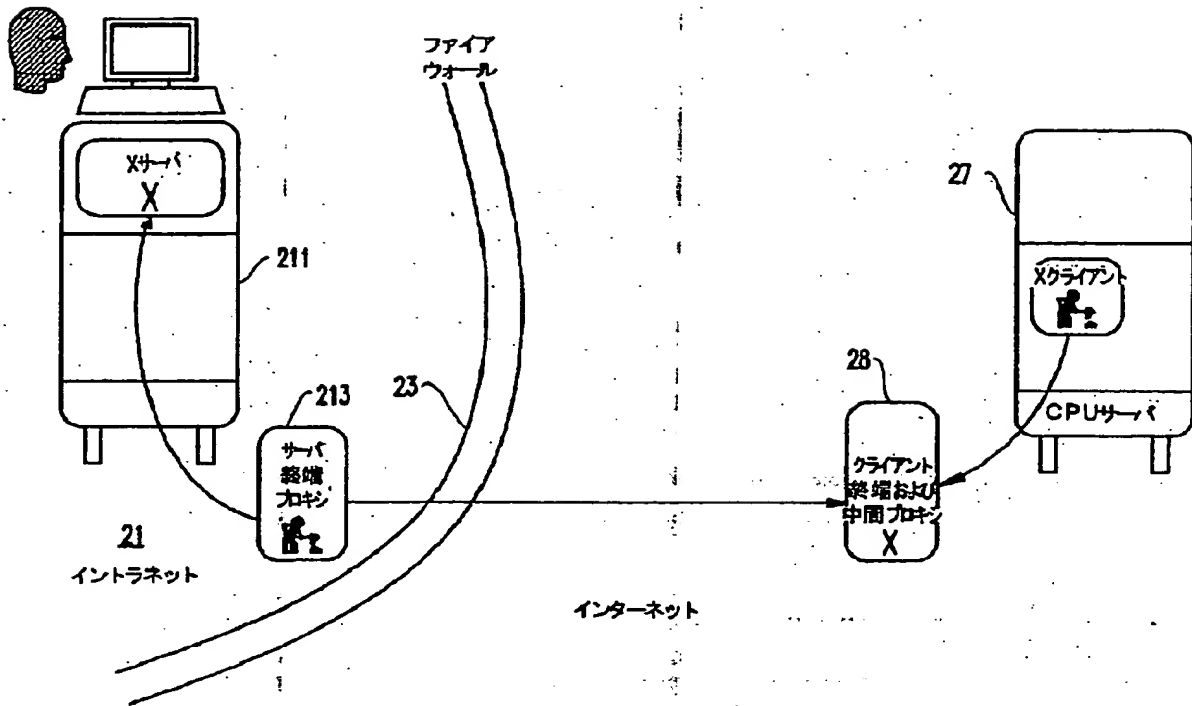




【図8】



【図9】





(72)発明者 ウィキ・エル・ライト  
アメリカ合衆国 94086 カリフォルニア  
州サニーベイル イースト・イブリン・ア  
ベニュー825 アpartment・ナンバー  
212